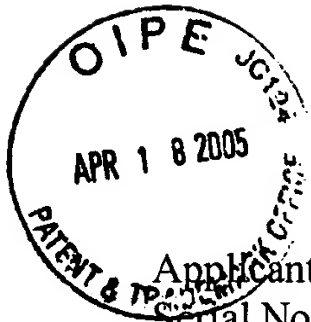# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Ravi P. Singh et al.   Art Unit   : 2183
Serial No. : 09/675,817   Examiner : Charles A. Harkness
Filed   : September 28, 2000
Title   : VARIABLE WIDTH INSTRUCTION ALIGNMENT ENGINE

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF ON BEHALF OF RAVI P. SINGH ET AL.

The Appeal Brief fee of **$500** is enclosed. Please apply any other charges or credits to Deposit Account No. 06-1050.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 2 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

(i.) Real Party in Interest

The real party in interest in the above application is Intel Corporation.

(ii.) Related Appeals and Interferences

The appellant is not aware of any appeals or interferences related to the above-identified patent application.

(iii.) Status of Claims

This is an appeal from the decision of the Primary Examiner in an Office Action dated November 19, 2004, rejecting claims 1, 3-8, 18-21, and 23-27, all of the claims of the above application. Claims 2, 9-17, and 22 were canceled. The claims have been twice rejected. Claims 1, 3-8, 18-21, and 23-27 are the subject of this appeal.

(iv.) Status of Amendments

All amendments have been entered. Appellant filed a Notice of Appeal on **February 18, 2005**.

(v.) Summary of Claimed Subject Matter

Background

The claimed invention relates to digital processors, and more particularly to alignment of instructions of variable bit widths within a digital processor. [Specification page 1, lines 2-4]

Appellant's Invention

Claim 1

One aspect of Appellant's invention is set out in claim 1, as a method of aligning instructions in a processor. Appellant's FIG. 5 shows a logic diagram of the data flow 500 in an alignment mux 615.

Claim 1 includes the feature of: "storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 3 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer." FIG. 5 shows that in the data flow 500, instructions are loaded into the memory 505. The memory 505 includes a plurality buffers 510, 515 for storing the instructions. Each of the "buffer areas" (buffer 510 and buffer 515) is divided into a plurality of smaller "sub-buffers." [Specification page 8, line 23 – page 9, line 7]
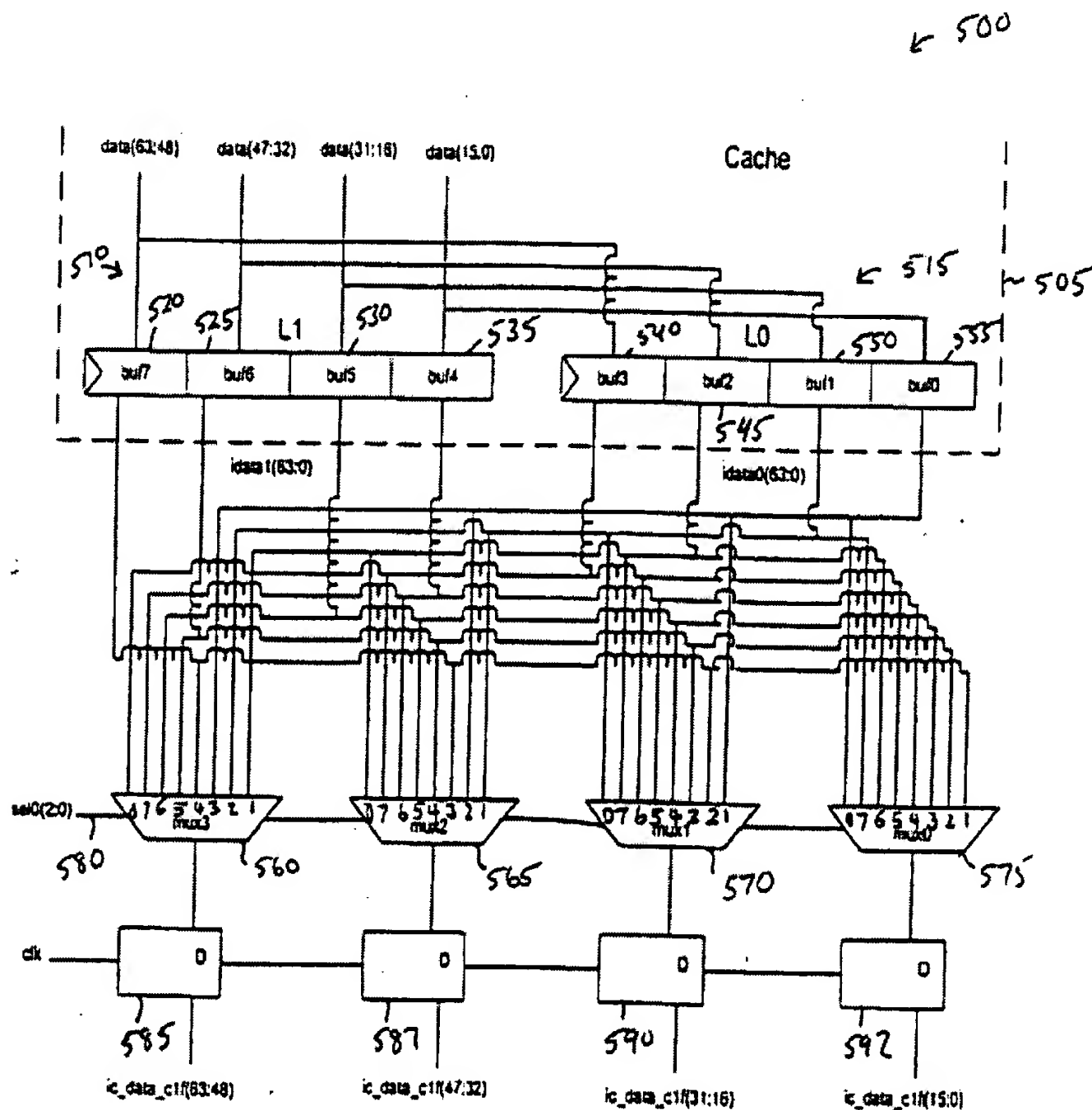

Figure 5

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 4 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

An individual instruction may be initially split among the plurality of 16-bit buffers 520-555 (where, in this embodiment, 16 bits is a "unit instruction width"). For example, a 64-bit instruction may begin in the third buffer 530 and end in the sixth buffer 545. [Specification page 10, lines 1 – 4]
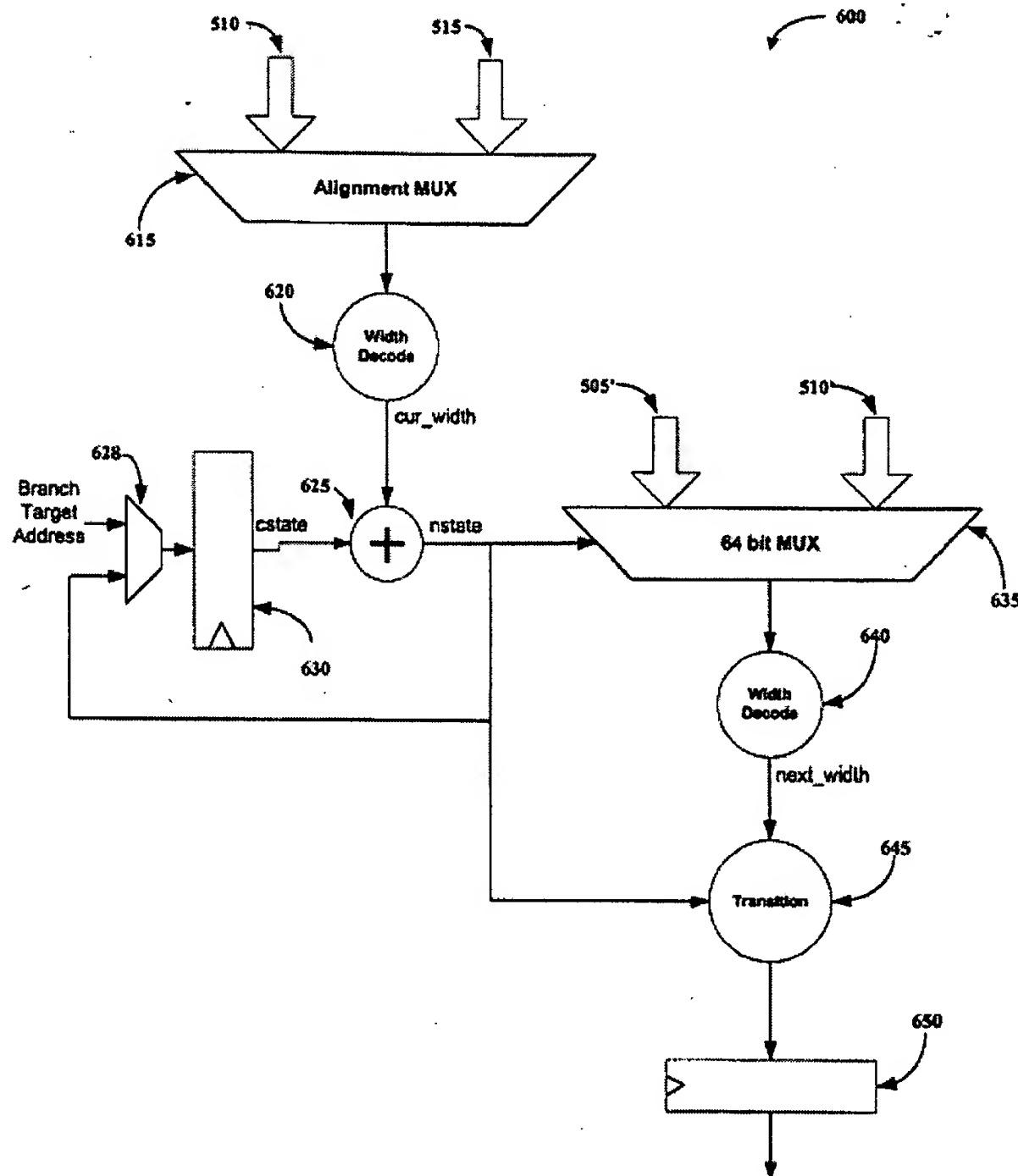

Figure 6

Appellant's claim 1 also includes the feature of: "aligning a first instruction from said buffer areas." Appellant's FIG. 6 shows a block diagram of an instruction request unit 600 including the alignment mux 615. A first instruction is aligned from the buffer areas. For example, the instruction request unit 600 receives instruction data from the buffers 510, 515

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 5 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

which are input into an alignment multiplexer 615. The alignment multiplexer 615 aligns the current instruction data received from the buffers 510, 515 and outputs the first instruction parsed from the instruction data. [Specification page 11, lines 13 – 18]

Appellant's claim 1 also includes the feature of: "decoding a size of the first instruction." The instruction request unit 600 decodes the size of the first instruction. When the first instruction is received from the alignment multiplexer 615, the first instruction is pre-decoded to generate width bits. The width of the first instruction (cur_width) is decoded in block 620 from width bits associated with the first instruction. [Specification page 11, line 19 – page 12, line 6]

Appellant's claim 1 also includes the feature of: "selecting at least one of said plurality of sub-buffers from which to output said first instruction on an output part." FIG. 5 shows each of the smaller, 16-bit buffers 520-555 connected to a plurality of selection multiplexers 560, 565, 570, and 575. Each of the selection multiplexers 560, 565, 570, 575 are connected to a select line 580 to select the output of the multiplexers 560, 565, 570, and 575. The multiplexers 560, 565, 570, and 575 each outputs a 16-bit signal which is stored in a plurality of flops 585, 587, 590, and 592 for use in the pipeline. [Specification page 9, lines 8 – 15]

Appellant's claim 1 also includes the feature of: "during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction." FIG. 6 shows a multiplexer 628 that receives a branch target address and the current state (cstate) of the instruction position in the buffers 510, 515. The multiplexer 628 selects either the branch target address or the current state to load the flop 630 with either the current state or the branch target address, which may become the current state. The current state is then combined with the width of the current instruction in block 625. By combining the current state with the current width, the position (nstate) of the beginning of the next instruction (the "second" instruction) may be determined. [Specification page 12, lines 7 – 16]

The instruction request unit 600 decodes the size of the second instruction. For example, the second alignment buffer 635 aligns the second instruction in the same manner the first alignment buffer 615 aligns the first instruction. The width of the second instruction is pre-decoded in block 640 to determine the corresponding width bits. The width information (next_width) is supplied to transition logic in block 645. [Specification page 13, lines 8 – 14]

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 6 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

Appellant's claim 1 also includes the features of "determining whether processing the second instruction will deplete said plurality of buffer areas," and "based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instructing the plurality of buffer areas to receive additional instructions."

The instruction request unit 600 determines whether processing the second instruction will deplete the plurality of buffer areas, and if so, instructs the plurality of buffer areas to receive additional instructions. For example, the transition logic determines whether either of the buffers 510 and 515 will be emptied after processing of the second instruction. The transition block 645 includes the next state position and the second instruction width as inputs. [Specification page 13, lines 15 – 18]

The transition block 645 determines based on the next state position and instruction width whether either of the buffers 510 and 515 will be exhausted after the second instruction. For example, if the next state position indicated is the beginning of the 16-bit buffer 530 and the second instruction width is 64-bits, the transition block 645 determines that the instruction will be taken from the 16-bit buffers 530, 535, 540, and 545, thus completely emptying the first buffer 510.

The transition block 645 may then send a signal to the flop 650 indicating that the first buffer 510 is available to be reloaded, which signals the memory to fill the empty buffer 510. [Specification page 13, line 15 – page 14, line 6]

Claim 21

Another aspect of the invention is covered by claim 21. Claim 21 is directed to an apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to align instructions in a processor. [Appellant's FIG. 6 shows a block diagram of a processor's instruction request unit 600 including an alignment mux 615.]

Claim 21 includes the feature that the system stores a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width. An instruction of greater than a unit instruction width is stored in more than one sub-buffer. [Specification page 8, line 23 – page 10, line 4]

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 7 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

Claim 21 also includes the feature that the system decodes a size of a first instruction from the plurality of buffer areas [Specification page 11, line 19 – page 12, line 6], and selects at least one of the plurality of sub-buffers from which to output the first instruction on an output part [Specification page 9, lines 8 – 15]. During the outputting, the system determines a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction [Specification page 12, lines 7 – 16].

Claim 21 also includes the feature that the system decodes the size of the second instruction, and determines whether processing the second instruction will deplete the plurality of buffer areas. Based on the determining whether processing the second instruction will deplete the plurality of buffer areas, the system instructs the plurality of buffer areas to receive additional instructions [Specification page 13, line 8 – page 14, line 6].

Claim 25

Another aspect of the invention is covered by claim 25. Claim 25 is a method of processing instructions within a processor. [Appellant's FIG. 6 shows a block diagram of a processor's instruction request unit 600 including an alignment mux 615.]

Claim 25 includes the feature of storing instructions of different widths within a cache having a plurality of buffer areas. Each buffer area has a plurality of subportions. Each subportion in the cache stores a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than the unit width takes up more than one subportion within the cache. [Specification page 8, line 23 – page 10, line 4]

Claim 25 also includes the feature of multiplexing each of the subportions of the cache to an output point [Specification page 9, lines 8 – 15], and selecting at least one of the cache subportions as a current instruction [Specification page 11, lines 15 – 18].

Claim 25 also includes the feature of, during the selecting the current instruction, predicting which of the buffer areas within the cache will be depleted of instruction data within a number of cycles approximately equal to a latency of the cache [Specification page 10, line 22 – page 11, line 9], and instructing loading of that number of buffer areas with additional instruction information [Specification page 13, line 15 – page 14, line 6].

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 8 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

Claim 27

Another aspect of the invention is covered by claim 27. Claim 27 is directed to a processor. [Appellant's FIG. 6 shows a block diagram of a processor's instruction request unit 600 including an alignment mux 615.]

Claim 27 includes the feature that the processor includes a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of the subparts storing a unit instruction width, and the instructions of greater than a unit instruction width being stored in multiple the subparts [Specification page 8, line 23 – page 10, line 4].

Claim 27 also includes the feature that the processor includes a multiplexer (e.g., alignment mux 615), connected to the plurality of subparts [Specification page 9, lines 8 – 15], and the multiplexer selects and aligns at least one of the plurality of subparts from any of the subparts within the buffer areas as a current instruction [Specification page 11, lines 15 – 18].

Claim 27 also includes the feature that the processor includes a predictor, operating to predict when at least one of the plurality of buffer areas will be empty, and to send a signal to instruct the at least one of the plurality of buffer areas to load another instruction data [Specification page 13, line 15 – page 14, line 6].

(vi.) Grounds of Rejection

(1) Claims 1, 3-6, 8, 21, and 23-24 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Zuraski, Jr. et al., U.S. Patent 6,260,134, and further in view of Nishii et al., U.S. Patent 5,918,045, and further in view of Narayan et al, U.S. Patent 6,161,172. Claims 1, 3-6, 8, and 21-24 are rejected in the summary of this rejection and claim 22 is mentioned in the body of the rejection. However, since claim 22 is indicated in the Office Action as no longer pending (claim 22 has been canceled), Appellant treats this rejection as a rejection of claims 1, 3-6, 8, 21, and 23-24.

(2) Claim 7 stands rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Zuraski '134, Nishii '045, and Narayan '172, and further in view of Davis, U.S. Patent 6,367,003.

(3)   Claims 18-19 and 25-27 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan '172, and further in view of Nishii '045. Claims 18-20 and 25-27 are rejected in the summary of this rejection. However, in the body of the rejection, the Examiner only rejects claims 18-19 and 25-27. Appellant treats this rejection as a rejection of claims 18-19 and 25-27.

(4)   Claim 20 stands rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Narayan '172, and Nishii '045, and further in view of Davis '003.

(vii.) Argument

Obviousness

"It is well established that the burden is on the PTO to establish a prima facie showing of obviousness," *In re Fritsch*, 972 F.2d. 1260, 23 U.S.P.Q.2d 1780 (C.C.P.A., 1972).

"It is well established that there must be some logical reason apparent from the evidence or record to justify combination or modification of references." *In re Regal*, 526 F.2d 1399 188, U.S.P.Q.2d 136 (C.C.P.A. 1975). "In addition, even if all of the elements of claims are disclosed in various prior art references, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art would have been prompted to combine the teachings of the references to arrive at the claimed invention. Id. Even if the cited references show the various elements suggested by the Examiner in order to support a conclusion that it would have been obvious to combine the cited references, the references must either expressly or impliedly suggest the claimed combination or the Examiner must present a convincing line of reasoning as to why one skilled in the art would have found the claimed invention obvious in light of the teachings of the references." *Ex Parte Clapp*, 227 U.S.P.Q.2d 972, 973 (Board. Pat. App. & Inf. 985).

"The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Gordon*, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984).

> Although the Commissioner suggests that [the structure in the primary prior art reference] could readily be modified to form the [claimed] structure, "[t]he mere fact that the prior art could be so

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 10 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Laskowski*, 10 U.S.P.Q. 2d 1397, 1398 (Fed. Cir. 1989).

"The claimed invention must be considered as a whole, and the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination." *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 488 (Fed. Cir. 1984).

> Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under Section 103, teachings of references can be combined only if there is some suggestion or incentive to do so. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984) (emphasis in original, footnotes omitted).

"The critical inquiry is whether 'there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.'" *Fromson v. Advance Offset Plate, Inc.*, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985).

"The concept of inherency is not applicable to the question of obviousness." *In re Sporman*, 363 F.2d 444, 150 USPQ 449 (CCPA 1965). To refer to an unexpected property or parameter as inherent begs the question of whether the unexpected property rebuts prima facie obviousness. The concept of inherency is not properly applicable to the question of obviousness (see, *In re Sporman*, 363 F.2d 444, 150 USPQ 449 (CCPA 1965)). Obviousness and inherency are entirely different questions; that which may be inherent is not necessarily known and, therefore, is an indication of unobviousness (*In re Sporman*, 363 F.2d 444, 449, 150 USPQ 449, 452 (CCPA 1965; see, also *In re Naylor*, 360 F.2d 765, 152 USPQ 106 (CCPA 1966); *In re Adams*, 356 F.2d 998, 148 USPQ 742 (CCPA 1966); and *In re Shetty*, 566 F.2d 81, 195 USPQ 753 (CCPA 1977)).

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 11 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

Discussion

**(1) Claims 1, 3-6, 8, 21, and 23-24 are patentably
distinct over Zuraski, Jr. et al., in view of Nishii
et al., and further in view of Narayan et al.**

<u>Claims 1, 4-6, 8, 21 and 24</u>

For the purposes of this appeal only, claims 1, 4-6, 8, 21 and 24 may be treated as standing or falling together. Claim 21 is representative of claims 1, 4-6, 8, 21 and 24.

Claim 21 recites "storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer." Claim 21 also recites instructions to "select at least one of said plurality of sub-buffers from which to output said first instruction on an output part; during said outputting, determining a beginning of a second instruction ***, decoding the size of the second instruction, and determining whether processing the second instruction will deplete the plurality of buffer areas; and based on said determining whether processing the second instruction will deplete the plurality of buffer areas, instruct the plurality of buffer areas to receive additional instructions."

The combination of these features is not taught by the combination of Zuraski, Nishii, and Narayan. The Examiner acknowledges (section 3 on page 3 of the 11-9-04 office action) that Zuraski does not teach:

> **receiving data containing instructions in a plurality of sub-buffers, selecting at least one of said buffer areas to output said first instruction on an output part; and determining whether processing the second instruction will deplete one of a plurality of buffer areas and instructing the plurality of buffer areas to receive additional instructions**

but contends (section 4 on page 3 of the 11-9-04 office action) that Nishii teaches:

> **determining whether processing the second instruction will deplete a buffers and instructing the buffer to receive additional data if processing the second instruction depletes the buffer**

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 12 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

In addition, the Examiner acknowledges (section 5 on page 4 of the 11-19-04 office action) that the combination of Zuraski and Nishii does not teach:

using a plurality of buffers, and receiving data containing instructions in a plurality of subbuffers, selecting at least one of said plurality of sub-buffer areas to output said first instruction on an output part

but contends (section 6 on page 4 of the 11-19-04 office action) that Narayan teaches:

receiving data containing instructions in a plurality of buffers, and selecting at least one of said buffer areas to output said first instruction on an output part

Assuming *arguendo* that a person of ordinary skill in the art was motivated to combine the teachings of Zuraski, Nishii and Narayan, Appellant contends that no proper combination of these references describes or suggests the claimed invention.

The Examiner's characterization of what Narayan teaches namely, receiving data containing instructions in a plurality of buffers" and that the subqueues 86A-C of Narayan's FIG. 4 correspond to Appellant's "plurality of buffers" is not correct. Instead, Narayan describes (column 19, line 65 – column 20, line 7):

According to one embodiment, instruction identification information is shifted internally to each subqueue 86 independently. Instruction identification information is not, therefore, shifted from position I0 of subqueue 86B into positions within subqueue 86A. Instead, when each of the instructions within subqueue 86A have been dispatched, subqueue 86B is shifted into subqueue 86A as a whole. The logic for shifting between subqueues 86 may operate independently from and in parallel with the internal shifting of each subqueue 86A-86C.

Narayan describes shifting "instruction identification information" among the subqueues. Narayan previously describes that "instruction identification information includes an indication of the validity of the instruction, as well as indications of the start and end of the instruction within the predefined number of instruction bytes" (column 6, lines 46 –50).

These teachings are contrary to the contentions of the examiner that Narayan teaches "receiving data containing instructions in a plurality of buffers" and that the subqueues 86A-C of Narayan's FIG. 4 corresponds to Appellant's "plurality of buffers." That is, not only does

Narayan fail to disclose "a plurality of buffer areas, each buffer area including a plurality of sub-buffers," but Narayan's description of shifting "instruction identification information" between subqueues is neither equivalent to nor suggests to "select at least one of said plurality of sub-buffers from which to output said first instruction on an output part."

On page 9 of the 11-19-04 office action, the Examiner states:

> Applicant has claimed a "unit instruction width" and an instruction that is larger than a unit instruction width. Any of the prior art could be applied to this since a unit instruction width is not defined, and could simply be a bit, or several bits in a queue or in the sub-queue in Narayan. Since there are instructions larger than the instruction width, as defined by the Applicant, it is not necessary for the instruction width to be the same size as the instructions. Therefore, the sub-queues of Narayan would have sub buffers, or instruction width sized sections, inside of each sub-queue in which an instruction would overlap many of the sub buffers.

The Examiner identifies subqueues 86A-C of Narayan's FIG. 4 as the "plurality of buffers" (section 6 on page 4 of the 11-19-04 office action). Then, in section 7 on page 4 of the same office action the Examiner inconsistently identifies the subqueues 86A-C of FIG. 4 as "a plurality of sub-buffers."

In addition, the Examiner suggests that the sub-queues of Narayan "would have sub buffers" even though no such sub-buffers are explicitly described in Narayan. Appellant contends that the Examiner is making an inherency argument and engaging in speculation of what is in the reference. Appellant further contends that this type of argument is improper since what a reference does or does not inherently contain has no bearing on the question of obviousness, as the case above clearly holds.

The Examiner must give patentable weight to all of the limitations of the claim. Neither Narayan, nor any combination of Narayan, Zuraski, and Nishi teaches "storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer," and to "select at least one of said plurality of sub-buffers from which to output said first instruction on an output part." Therefore, the combination of references cannot render claim 21 obvious, since they fail to suggest every element of claim 21.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 14 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

Furthermore, Appellant's claim 21 is distinct over the combination of Zuraski, Nishii and Narayan because, the Examiner has not presented a cogent line of reasoning as to why one skilled in the art, having the references before him, would modify and combine the teachings of Zuraski, Nishii and Narayan as set out by the examiner.

The Examiner states that "Nishii always keeps the prefetch buffer from being empty <u>by comparing the two pointers</u> and determining when the buffer needs to have more instructions fetched from memory" (relying on column 8, lines 38-60 of Nishii). Indeed, Nishii does describe prefetching instructions so that the buffer always holds instructions, and storing an instruction <u>based on the value of the write pointer</u>. Nishii at column 8, lines 45-57 states:

> **The control logic unit 518 receives enumerated values provided by the read pointer 510 and the write pointer 511 to detect the full status and empty status of the FIFO buffer 101, and prefetches instructions from the cache memory unit 6, so that the FIFO buffer 101 always holds instructions. The control logic unit 518 controls the prefetch pointer 512 to output the instruction address 143 for prefetching. In synchronism with this output, the control logic unit 518 sends a request signal 121 for accessing an instruction to the cache memory 6. The control logic unit 518 then waits for a response from the cache memory unit 6 and causes the FIFO buffer 101 to store an instruction based on the value of the write pointer 511.**

Unlike the arrangement taught by Nishii, where Nishii teaches receiving additional instructions merely based on values of pointers, Appellant's claim 21 requires "determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction, determining whether processing the second instruction will deplete the plurality of buffer areas; and <u>based on said determining</u> ***, instruct the plurality of buffer areas to receive additional instructions."

Nishii teachings are directed to a processor, where presumably all instructions are of the same width. None of the references, nor the reasoning offered by the examiner would motivate one of ordinary skill to modify Nishii's teaching of "fetching a branch target instruction in a data processor which is capable of prefetching an instruction" (column 1, lines 5-7) and modify that teaching to determine whether processing the second instruction will deplete the plurality of buffer areas; and based on said determining ***, instruct the plurality of buffer areas to receive additional instructions.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 15 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

The Examiner's stated motivation for combining Zuraski and Nishii with the teachings of Narayan, is: "having a plurality of buffers [allowing] the system to execute in parallel *** to reduce the amount of time required for execution." Appellant contends that this motivation is irrelevant to Appellant's claims. One would not be motivated by "allowing the system to execute in parallel" to modify Narayan to: "[store] a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer," as required by claim 21. Again, the Examiner has not identified a teaching, suggestion, or motivation in any of the references as a basis to modify Narayan, or presented a cogent line of reasoning that would have suggested such modifications to those of ordinary skill in the art.

Claims 3 and 23

For the purposes of this appeal only, claims 3 and 23 may be treated as standing or falling together. Claim 3 is representative of claims 3 and 23.

Claim 3 recites "comparing a most significant bit of a pointer to a first sub-buffer to a most significant bit of a pointer to a second sub-buffer to determine whether processing one of the plurality of instructions will deplete any of the buffer areas." The Examiner states on page 5 of the 11-19-04 office action that Nishii teaches this at column 8, lines 38-60. However, at column 8, lines 38-60 Nishii teaches:

> A control logic unit (CLOG) 518 controls the pointers 510-512 and controls access to the cache memory unit 6. The control logic unit 518 receives the initializing signal 516, clearance blocking signal 132, request signal 142 for fetching instructions from the timing controller 141, acknowledge signal 123 from the cache memory unit 6, and enumerated values by the read pointer 510 and the write pointer 511. The control logic unit 518 receives enumerated values provided by the read pointer 510 and the write pointer 511 to detect the full status and empty status of the FIFO buffer 101, and prefetches instructions from the cache memory unit 6, so that the FIFO buffer 101 always holds instructions. The control logic unit 518 controls the prefetch pointer 512 to output the instruction address 143 for prefetching. In synchronism with this output, the control logic unit 518 sends a request signal 121 for accessing an instruction to the cache memory 6. The control logic unit 518 then waits for a response from the cache memory unit 6 and causes the FIFO buffer 101 to store an instruction based on the value of the write pointer 511. And, each time the FIFO buffer 101 stores an instruction,

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 16 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

> the control logic unit 518 controls the prefetch pointer 512 and the write
> pointer 511 to be incremented by a write clock signal 517.

This teaching in Nishii does not suggest "comparing a most significant bit" of one pointer with that of another, as required by claim 3. Neither in the quoted portion of Nishii nor elsewhere in Nishii, Narayan or Zuraski, is the recited limitation suggested.

(2) Claim 7 is patentably distinct over the combination of Zuraski, Nishii, and Narayan, and further in view of Davis.

<u>Claim 7</u>

Claim 7 is patentable for at least the same reasons as claim 1.

(3) Claims 18-19 and 25-27 are patentably distinct over Narayan, and further in view of Nishii.

<u>Claim 25</u>

As argued above for claim 21, the Examiner has not presented a cogent line of reasoning as to why one skilled in the art, having the references before him, would modify and combine the teachings of Nishii and Narayan. Nevertheless, even if one were to combine the teachings of Nishii and Narayan, no proper combination would teach or suggest the claimed invention.

Claim 25 recites "storing instructions of different widths within a cache having a plurality of buffer areas, each buffer area having a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache." Claim 25 also recites "selecting at least one of said cache subportions as a current instruction."

The Examiner states on page 6 of the 11-19-04 office action that Narayan teaches "a processor comprising: a plurality of buffer areas (Narayan '172 figure 4 reference numbers 86A-C), adapted to store a plurality of instructions of different width in a plurality of subparts, each of said subparts storing a unit instruction width, and said instructions of greater than unit instruction widths being stored in multiple said subparts (Narayan '172 column 6, lines 23-65)." On page 7 the Examiner further states that Narayan teaches "a multiplexor, connected to said plurality of

Applicant : Ravi P. Singh et al.   Attorney's Docket No.: 10559-284001 / P9291 - ADI
Serial No. : 09/675,817   APD1803-1-US
Filed : September 28, 2000
Page : 17 of 24

subparts, and selecting and aligning one of said plurality of subparts from any of said subparts within said buffer areas as a current instruction (Narayan '172 figures 4 and 6, column 20, lines 18-33, column 23, lines 3-27)."

The Examiner's characterization of what Narayan teaches is not correct. As in the argument above for claim 21, if Narayan's subqueues 86A-C are interpreted as the recited plurality of buffer areas, then Narayan does not teach or suggest that each subqueue has "a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache."

The passage in Narayan cited by the Examiner (column 6, lines 23-65) as allegedly relating to storing an instruction in one or more subparts instead describes:

> **Microprocessor 10 is configured to align instructions from instruction cache 16 to decode units 20 using instruction alignment unit 18. Instructions are fetched as an aligned plurality of bytes from a cache line within instruction cache 16. Instructions of interest may be stored beginning at any arbitrary byte within the fetched bytes. For example, a branch instruction may be executed having a target address which lies within a cache line. The instructions of interest therefore begin at the byte identified by the target address of the branch instruction. From the instruction bytes fetched, instruction alignment unit 18 identifies the instructions to be executed. Instruction alignment unit 18 conveys the instructions, in predicted program order, to decode units 20 for decode and execution.**

Narayan in this passage states that: "instructions are fetched as an aligned plurality of bytes" and that "from the instruction bytes fetched, instruction alignment unit 18 identifies instructions to be executed." Narayan does not describe anything that can be interpreted as a subportion of a subqueue, much less that an instruction of more than unit width takes up more than one subportion.

To the contrary, Narayan describes the subqueues as having "instruction positions" that store "instruction information." For example, in column 19, lines 18 – 37 Narayan describes:

> **Subqueues 86 store instruction information in a plurality of instruction positions (or simply "positions"). The number of instruction positions is preferably equal to the maximum number of instructions which may be included in an instruction block. For the present embodiment, three positions are included. The first position ("position I0") stores the**

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 18 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

> **instruction identification information corresponding to the instruction which is foremost in program order within the instruction block stored in the subqueue 86. The second position ("position I1") stores the instruction identification information corresponding to the second instruction in program order within the instruction block. Finally, the third position ("position I2") stores the instruction identification information corresponding to the last instruction in program order. Alternatively, position I2 may store instruction identification information corresponding to an overflow instruction. Certain instruction identification information is the same for each instruction (e.g. the segment limit). To avoid duplicating information, this instruction information may be stored as a single copy separate from the instructions positions.**

This passage identifies "instruction information" as the "instruction identification information" previously defined (column 6, lines 46 –50) as "instruction identification information includes an indication of the validity of the instruction, as well as indications of the start and end of the instruction within the predefined number of instruction bytes."

Thus, even if Narayan's subqueues are interpreted as a plurality of buffer areas, Narayan does not describe a subqueue as having "a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache," or "selecting at least one of said cache subportions as a current instruction." Likewise Nishii does not teach or suggest these limitations of claim 25. Therefore, claim 25 is patentably distinct over Narayan in view of Nishii.

Claim 26

Claim 26 recites that "predicting comprises comparing a most significant bit of a pointer to a first subportion, to a most significant bit of a pointer to a second subportion, to determine if any of the buffer areas will be depleted." The Examiner states on pages 7 – 8 of the 11-19-04 office action that Nishii teaches this at column 8, lines 38-60.

However, as argued above, this teaching in Nishii does not suggest comparing "a most significant bit" of one pointer with that of another, as required by claim 18. Neither Nishii nor Narayan, alone or in combination, teach or suggest the recited limitation.

Claims 19 and 27

For the purposes of this appeal only, claims 19 and 27 may be treated as standing or falling together. Claim 27 is representative of claims 19 and 27.

Claim 27 recites "a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width, and said instructions of greater than a unit instruction width being stored in multiple said subparts," and "a multiplexer, connected to said plurality of subparts, and selecting and aligning at least one of said plurality of subparts from any of said subparts within said buffer areas as a current instruction."

As in the argument above for claim 25, even if Narayan's subqueues are interpreted as a plurality of buffer areas, Narayan does not describe a subqueue "adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width, and said instructions of greater than a unit instruction width being stored in multiple said subparts," or "selecting and aligning at least one of said plurality of subparts from any of said subparts within said buffer areas as a current instruction." Neither does Nishii teach or suggest these limitations of claim 27. Therefore, claim 27 is patentably distinct over Narayan in view of Nishii.

Claim 18

Claim 18 recites that "the transition detector compares a most significant bit of a pointer to a first subpart to a most significant bit of a pointer of a second subpart to determine whether processing one of the plurality of instructions will deplete any of the buffer areas." The Examiner states on page 8 of the 11-19-04 office action that Nishii teaches this at column 8, lines 38-60.

However, as argued above, this teaching in Nishii does not suggest comparing "a most significant bit" of one pointer with that of another, as required by claim 18. Neither Nishii nor Narayan, alone or in combination, teach or suggest the recited limitation.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 20 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

(4) Claim 20 is patentably distinct over the combination of Narayan, and Nishii, and further in view of Davis.
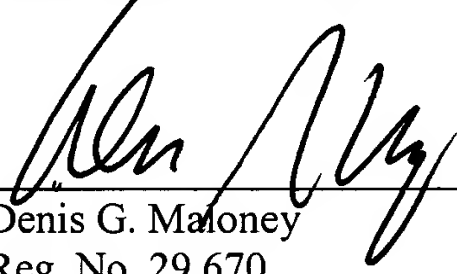
Claim 20

Claim 20 is patentable for at least the same reasons as claim 27.

Conclusion

Appellant submits, therefore, that claims 1, 3-8, 18-21, and 23-27 are allowable over the cited art. Therefore, the Examiner erred in rejecting Appellant's claims and should be reversed.

Respectfully submitted,

Date: April 15, 2005

Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21017127.doc

Applicant : Ravi P. Singh et al.   Attorney's Docket No.: 10559-284001 / P9291 - ADI
Serial No. : 09/675,817   APD1803-1-US
Filed : September 28, 2000
Page : 21 of 24

Appendix of Claims

1.   A method of aligning instructions in a processor comprising:

storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer;

aligning a first instruction from said buffer areas;

decoding a size of the first instruction;

selecting at least one of said plurality of sub-buffers from which to output said first instruction on an output part;

during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction, and determining whether processing the second instruction will deplete said plurality of buffer areas; and

based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instructing the plurality of buffer areas to receive additional instructions.

Claim 2 is canceled.

3.   The method of Claim 1, further comprising comparing a most significant bit of a pointer to a first sub-buffer to a most significant bit of a pointer to a second sub-buffer to determine whether processing one of the plurality of instructions will deplete any of the buffer areas.

4.   The method of Claim 1, further comprising storing a first instruction across a plurality of sub-buffers prior to processing the instructions.

5.   The method of Claim 1, further comprising adding the size of the first instruction to a current instruction position to determine the beginning of the second instruction.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 22 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

6.   The method of Claim 1, further comprising aligning ahead a number of cycles equal to a cache latency.

7.   The method of Claim 1, further comprising aligning instructions in a digital signal processor.

8.   The method of Claim 1, further comprising issuing a request to a memory to reload the plurality of buffer areas.

Claims 9-17 are canceled.

18.   The processor of Claim 27, wherein the transition detector compares a most significant bit of a pointer to a first subpart to a most significant bit of a pointer of a second subpart to determine whether processing one of the plurality of instructions will deplete any of the buffer areas.

19.   The processor of Claim 27, wherein the processor aligns ahead a number of cycles equal to a cache latency.

20.   The processor of Claim 27, wherein the processor is a digital signal processor.

21.   An apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to align instructions in a processor, the instructions causing the machine to:

storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer;

decode a size of a first instruction from said plurality of buffer areas;

select at least one of said plurality of sub-buffers from which to output said first instruction on an output part;

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 23 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction, and determining whether processing the second instruction will deplete the plurality of buffer areas; and

based on said determining whether processing the second instruction will deplete the plurality of buffer areas, instruct the plurality of buffer areas to receive additional instructions.

Claim 22 is canceled.

23.   The apparatus of Claim 21, wherein a most significant bit of a pointer to a first sub-buffer is compared to a most significant bit of pointer to a second sub-buffer to determine whether processing one of the plurality of instructions will deplete any of the buffer areas.

24.   The apparatus of Claim 21, wherein a first instruction is stored across a plurality of sub-buffers prior to processing the instructions.

25.   A method of processing instructions within a processor, comprising:

storing instructions of different widths within a cache having a plurality of buffer areas, each buffer area having a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache;

multiplexing each of the subportions of said cache to an output point, and selecting at least one of said cache subportions as a current instruction;

during said selecting said current instruction, predicting which of said buffer areas within said cache will be depleted of instruction data within a number of cycles approximately equal to a latency of the cache, and instructing loading of that number of buffer areas with additional instruction information.

Applicant : Ravi P. Singh et al.
Serial No. : 09/675,817
Filed : September 28, 2000
Page : 24 of 24

Attorney's Docket No.: 10559-284001 / P9291 - ADI
APD1803-1-US

26. A method as in claim 25, wherein said predicting comprises comparing a most significant bit of a pointer to a first subportion, to a most significant bit of a pointer to a second subportion, to determine if any of the buffer areas will be depleted.

27. A processor comprising:

a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width, and said instructions of greater than a unit instruction width being stored in multiple said subparts;

a multiplexer, connected to said plurality of subparts, and selecting and aligning at least one of said plurality of subparts from any of said subparts within said buffer areas as a current instruction; and

a predictor, operating to predict when at least one of the plurality of buffer areas will be empty, and to send a signal to instruct said at least one of the plurality of buffer areas to load another instruction data.